

## **IN THE SPECIFICATION**

Please amend the specification as follows:

Please replace the paragraph beginning on page 1, line 7, with the following rewritten paragraph as follows:

-- When calling up an object, it becomes necessary to know where the object to be called up exists. As one of the methods for acquiring information on a location at which the object exists or the like, as disclosed at <http://www.omg.org>, there exists a method of utilizing a naming service. Incidentally, basically the same technology has been disclosed in <http://www.iona.com/support/docs/manuals/orbix/32/html/orbixnames33-pguide/LoadBalancingC++.html> as well. --

Please replace the paragraph beginning on page 2, line 16, with the following rewritten paragraph as follows:

-- The naming service stores therein object names and object reference information. In the present invention, the object names, which refer to logical names of objects to be registered, are names for uniquely identifying the objects in a system. Also, each of the object reference information, which is an identifier for specifying the existence location of each object resource, is indicated by a combination of an IP address, a port number, a URL, and the like. According to the information disclosed at <http://www.omg.org>, the object reference information is also referred to as "object reference". In the present invention, a combination of each object name, each object reference information, and the like is referred to as "each entry". Each entry can be retrieved and acquired using the corresponding object name. Usually, the

naming service performs the retrieval using the object name as the key, thereby acquiring the object reference information corresponding thereto. Then, taking advantage of this information, the object is requested to perform a processing. This processing request to the object causes communications therefor to occur. --

Please replace the paragraph beginning on page 8, line 15, with the following rewritten paragraph as follows:

-- Each container (110, 210) provides the execution environment of an object. Each object (111, 112, 113, 211, 212, 213) executes a processing requested thereto. Each naming interface unit (120, 220) is an interface used when accessing the naming service unit 310. Each cache unit (130, 230) stores the object name and the object reference information of a notified object. Each cache unit is divided into each local cache and each global cache. Each remote calling-up unit (140, 240) is of a communication layer for sending a request to a remote object. As a publicly-known example, there exists a technology indicated at <http://java.sun.com>. The present embodiment has used functions based thereon. Each distributed-object base unit (150, 250, 350) is of a communication layer for sending a request to a distributed object. The technology disclosed at <http://java.sun.com> has been applied with no change added thereto. --